

BECA: A Software Tool for Integrated Visualization of Human Brain Data

Huang Li^{1,2}, Shiaofen Fang^{1*}, Bob Zigon¹, Olaf Sporns³, Andrew J. Saykin², Joaquín Goñi^{4,5,6}, Li Shen^{1,2}

¹ Computer and Information Science, Purdue University Indianapolis, IN, USA

² Radiology and Imaging Sciences, Indiana University School of Medicine, IN, USA

³ Psychological and Brain Sciences, Indiana University Bloomington, IN, USA

⁴ School of Industrial Engineering, Purdue University, West-Lafayette, IN, USA

⁵ Weldon School of Biomedical Engineering, Purdue University, West-Lafayette, IN, USA

⁶ Purdue Institute for Integrative Neuroscience, Purdue University, West-Lafayette, IN, USA

Abstract. Visualization plays an important role in helping neuroscientist understanding human brain data. Most publicly available software focuses on visualizing a specific brain imaging modality. Here we present an extensible visualization platform, BECA, which employ a plugin architecture to facilitate rapid development and deployment of visualization for human brain data. This paper will introduce the architecture and discuss some important design decisions in implementing the BECA platform and its visualization plugins.

Keywords: visualization, human brain data, plugin platform, software architecture

1 Introduction

The advancement of medical imaging technologies produces a large amount of neuroimaging data which is capable of showing different brain properties. In addition, a number of measurements are calculated along with the raw data. To give neuroscientist an intuitive understanding of the raw data and their measurements, visualization plays a vital role. There arises the need to design and build a visualization software tool to visualize all kinds of raw data and measurements in the context of their spatial structures.

In collaboration with neuroscientist, four types of brain imaging and genomics data are integratively employed and visualized in the software: diffusion tensor imaging (DTI) fiber tracts, structural magnetic resonance imaging (sMRI) data, functional magnetic resonance imaging (fMRI) data, and genomic data.

A few visualization software tools have been released for visualizing neuroimaging data. But they often focus on visualizing few types of brain data. For example, MRICron [1] is developed to show 2D slice of sMRI or fMRI data. TrackVis [2] can be used to

* Corresponding author: Shiaofen Fang (shfang@iupui.edu)

visualize DTI fiber tracts and sMRI data. FSLview [3] is capable of visualizing sMRI and fMRI data. To facilitate an interactive exploration of multimodal brain image data, it is of high value to build a framework that can visualize more imaging modalities.

To address this issue, we present BECA, an extensible visualization platform that accepts plugins for visualizing different types of human brain data. Those types are managed by different plugins which share the same platform.

2 Design and Implementation

The challenge of designing BECA comes from two aspects: firstly, different dataset requires different visualization methods. Even for just one dataset, domain expert may want to use different visualization techniques in different occasions. For example, for the a sMRI dataset, we can use either volume rendering to visualize MRI dataset for voxel-based analysis or isosurface rendering for surface analysis. On the other hand, from the view of software engineering, in the process of iterative software development, we need to deliver the software periodically in order to retrieve feedbacks from domain experts. Therefore, we introduce a plugin architecture which greatly facilitates the development and deployment of BECA.

2.1 Architecture

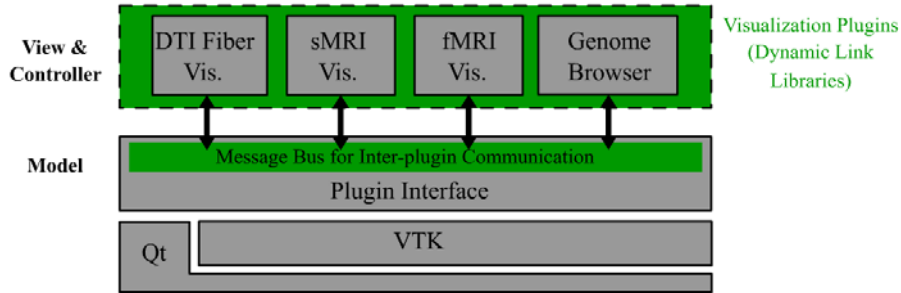


Fig. 1. Software architecture of BECA.

An overview of the software architecture is shown in Figure 1. The Qt library [4] is used to build the user interface, and VTK [5] is used for 3D visualization. The plugin architecture is based on the classic Model-View-Controller (MVC) [6] design pattern. The Model component holds the data from the application domain, such as sMRI data as 3D image or fMRI data as 4D image, which are rarely changed in the development of the BECA. In contrast, the View and Controller components are much more unstable because they are responsible for representing the visualizations and responding to user interactions which vary from one visualization to another. Furthermore, in the iterative software development model, the visualization and interaction are very likely to be changed based on the feedbacks from neuroscientists. Therefore, we designed the plugin architecture to decouple the unstable View and Controller from the stable Model

component. Different visualization plugins have different View and Controller component, but they share the same Model component. With the plugin architecture, each visualization plugin can be added or modified easily without changing the underlying infrastructure or other plugins. We also implemented a message bus for message passing between different visualization plugins, so that plugins can communicate with each other for implementing functions such as synchronizing the views in different visualizations. We have released the message bus as an open-source project hosted on Github (<https://github.com/lheric/libgitlevtbus>).

2.2 DTI Tractography Visualization

The pipeline for visualizing DTI tracts is shown in Figure 2. Tracts are accessed via the plugin interface. We write a fiberColorFilter to color each fiber tract by its own property such as direction, length, or distance between endpoints.

Suppose a tract consist of $N + 1$ points $P_0(x_0, y_0, z_0), P_1(x_1, y_1, z_1), \dots, P_n(x_n, y_n, z_n)$. The direction is encoded as a RGB color by following formula:

$$R = |x_0 - x_n|/d(P_0, P_n)$$

$$G = |y_0 - y_n|/d(P_0, P_n)$$

$$B = |z_0 - z_n|/d(P_0, P_n)$$

where function $d(p, q)$ computes the Euclidean distance between p and q . The length or distance of a tract is a scalar value which can be mapped into color via a lookup table. An example result is shown in Figure 3.

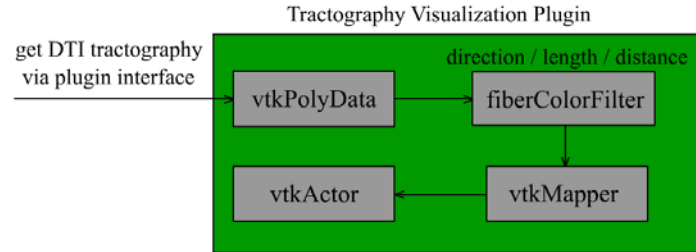


Fig. 2. VTK pipeline for visualizing DTI tractography

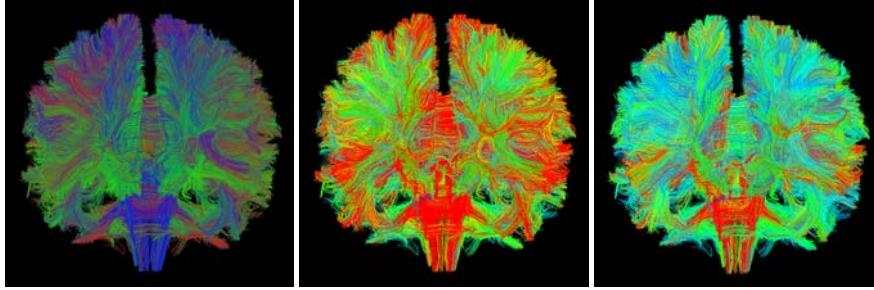


Fig. 3. Map direction (left), length (middle), and distance (right) as tract color.

2.3 sMRI Visualization

For sMRI visualization, the pipeline is shown in Figure 4. The visualization plugin takes sMRI data as a 3D image. Then each region of interest (ROI) is extracted with `vtkDiscreteMarchingCubes` class in the VTK library. Each ROI get its own `vtkActor` so that they can have different colors or textures, as shown in Figure 5.

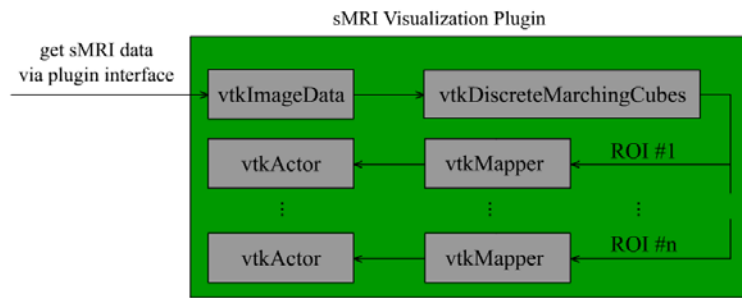


Fig. 4. VTK pipeline for visualizing sMRI



Fig. 5. Result of sMRI visualization.

2.4 fMRI Visualization

We use the texture mapping approach to visualize fMRI data on their corresponding ROI surfaces. We need to first encode this time-series data on a 2D texture image. We propose an offset contour method to generate patterns of contours based on the boundary of each projected ROI. The offset contours are generated by offsetting the boundary curve toward the interior of the region, creating multiple offset boundary curves. The technical detail is depicted in [7].

The implementation of the fMRI visualization pipeline is very similar to that in sMRI visualization, except that `vtkTexture` class to map the generated texture onto the surface of each ROI. Figure 6 shows an example output of fMRI visualization.

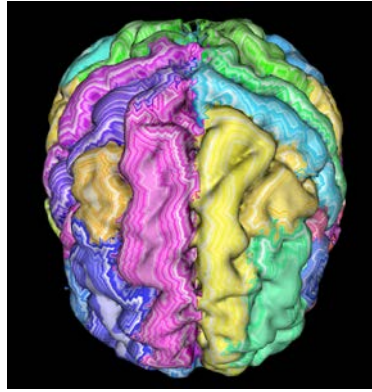


Fig. 6. An example of fMRI data visualization

2.5 GPU Accelerated Genome Browser

In BECA, we also implemented a browser for neuroimaging genomic data that implemented analysis of variance (ANOVA statistical test) and genome-wide association studies (GWAS). Different from previous visualization, the genome browser is computational intensive, which require GPU acceleration for real-time interaction. We used the server-client model in the genome browser, as shown in Figure 7. The server is responsible for computation using CUDA [8] with NVIDIA GPUs, while the client is for display. The client and server communicate through a TCP/IP socket. There are mainly two benefits by employing the server-client model to decouple visualization from computation: firstly, the client can run on a terminal without a NVIDIA graphics card, which lower the hardware requirement to run the genome browser; secondary, different clients can share the same server, which further reduce the cost on deployment.

Figure 8 shows the primary components of the user interface. In the lower left hand corner, a 3-dimensional model of a reference brain is displayed, color-mapped with the p-value of the association between each voxel and the current SNP or gene. At the top of the user interface is the SNP or gene explorer. This region displays the $-\log_{10}(p\text{-value})$ of the association between each SNP or gene and the current voxel.

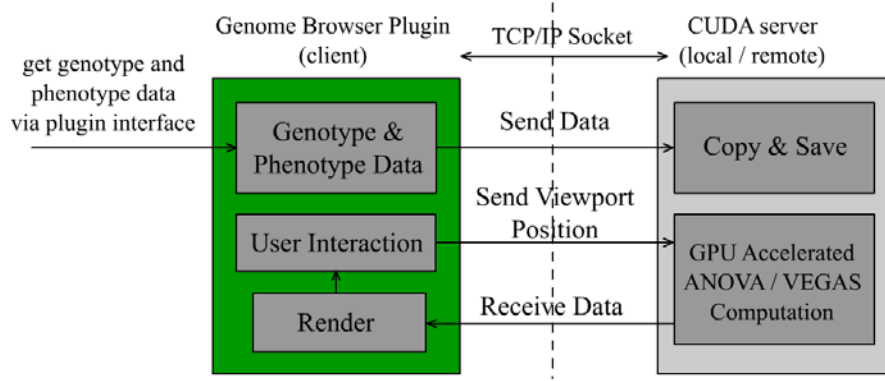


Fig. 7. Client-server model for GPU accelerated genome browser

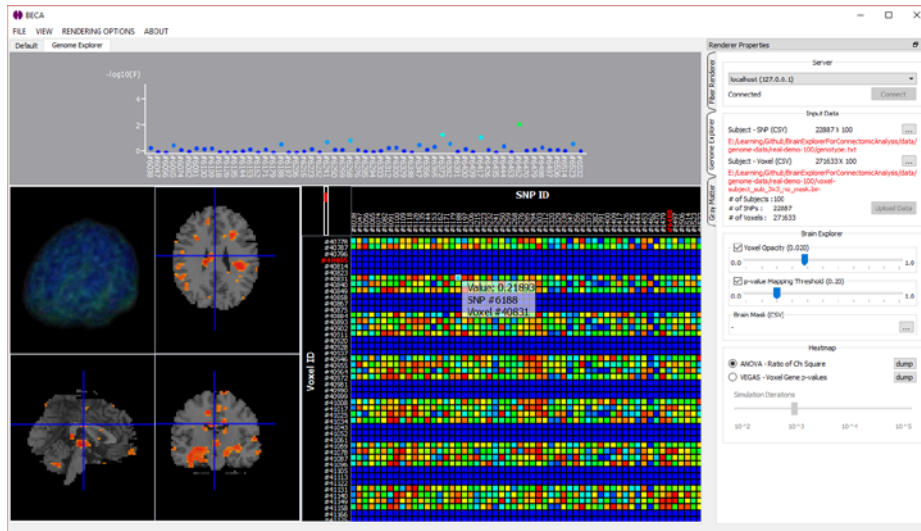


Fig. 8. User interface for the neuroimaging genome browser, which consist of SNP explorer (top), brain explorer (bottom left), and heat map (bottom right).

3 Conclusion

In this paper, we presented the design of our visualization platform, BECA, for integrated visualization of human brain data. The software is freely available at <http://www.iu.edu/~beca/>. It provided a plugin platform for rapid implementation of different visualizations for all kinds of brain data. In corporation with domain experts of neuroscience, work is still in progress to support more data formats and visualization techniques.

Acknowledgements

This work was supported by NIH R01 EB022574, R01 LM011360, U01 AG024904, RC2 AG036535, R01 AG19771, P30 AG10133, UL1 TR001108, R01 AG 042437, and R01 AG046171; DOD W81XWH-14-2-0151, W81XWH-13-1-0259, and W81XWH-12-2-0012; NCAA 14132004; and IUPUI ITDP Program.

References

1. Wang R., Van J.: Diffusion Toolkit: A Software Package for Diffusion Imaging Data Processing and Tractography. *Proc. Intl. Soc. Mag. Reson. Med.* 15 (2007)
2. Neuropsychology Lab: MRICron. <http://people.cas.sc.edu/rorden/mricron/index.html>. Accessed 30 July 2017
3. Jenkinson M. et al.: FSL. *NeuroImage*, 62, 782-90 (2012)
4. The Qt Company: Qt Framework. <https://www.qt.io>. Accessed 30 July 2017
5. Kitware: The Visualization Toolkit. <http://www.vtk.org>. Accessed 30 July 2017
6. Fowler M.: *Patterns of Enterprise Application Architecture*, Addison-Wesley Professional, (2003)
7. Li H., Fang S., Cortes GJ., Contreras J., Liang L., Cai C., West J., Risacher S., Wang Y., Sporns O., Saykin A., Shen L.: for the ADNI Integrated Visualization of Human Brain Connectome Data, *Brain Informatics and Health*: Springer International Publishing, 295-305 (2015)
8. Nickolls J.: Scalable Parallel Programming with CUDA. *Queue* 6, 2 (2008)